



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10228417 A**(43) Date of publication of application: **25.08.98**

(51) Int. Cl

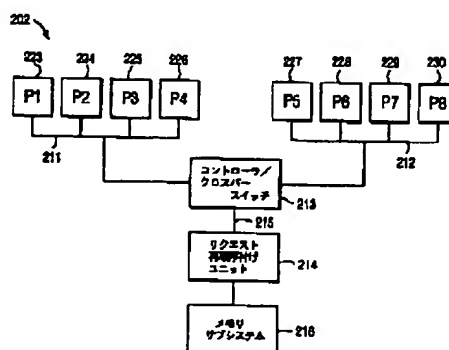
G06F 12/06(21) Application number: **09311893**(22) Date of filing: **13.11.97**(30) Priority: **15.11.96 US 96 31063**
28.02.97 US 97 808849(71) Applicant: **HYUNDAI ELECTRON AMERICA
INC**(72) Inventor: **TALBOT GERRY R**
HYPHER AUSTEN J(54) **DATA PROCESSOR**

(57) Abstract:

PROBLEM TO BE SOLVED: To control an access to a memory subsystem to maximize bandwidth and parallel property by selecting a request that accesses a usable memory position among plural requests and scheduling the selected request.

SOLUTION: Processors P1(223) to P4(226) generate a request for a memory subsystem 216 to a 1st memory bus 211. Processors P5(227) to P8(230) generates a request for the subsystem 216 to a 2nd memory bus 212. A controller/crossbar switch 213 makes routing of a memory request from the two memory buses 211 and 212 to a 3rd bus 215. A request resequencing unit 214 accepts a request from the bus 215 and schedules the request. The sequence to be scheduled is controlled so that a memory access request may perform service.

COPYRIGHT: (C)1998,JPO



THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-228417

(43) 公開日 平成10年(1998) 8月25日

(51) Int.Cl.⁴

G 0 6 F 12/06

識別記号

5 5 0

F I

G 0 6 F 12/06

5 5 0 A

審査請求 未請求 請求項の数17 OL (全 17 頁)

(21) 出願番号 特願平9-311893

(22) 出願日 平成9年(1997)11月13日

(31) 優先権主張番号 60/031063

(32) 優先日 1996年11月15日

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 08/808849

(32) 優先日 1997年2月28日

(33) 優先権主張国 米国 (U S)

(71) 出願人 591020870

ヒュンダイ エレクトロニクス アメリカ
HYUNDAI ELECTRONICS
AMERICA

アメリカ合衆国, カリフォルニア
95134, サン ノゼ, ベイポイント パ
ークウェイ 166

(72) 発明者 ゲーリー アール タルボット

アメリカ合衆国 マサチューセッツ州
01742-4915 コンコード ホールデンウ
ッド ロード 280

(74) 代理人 弁理士 中村 稔 (外6名)

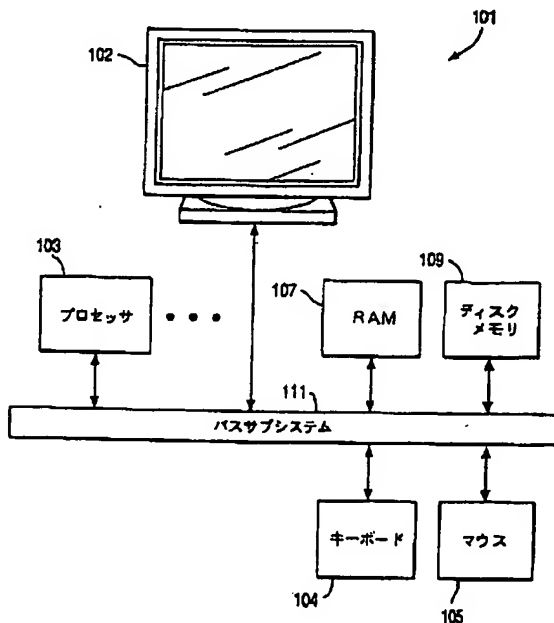
最終頁に続く

(54) 【発明の名称】 データ処理システムにおけるメモリリクエスト再順序付け法

(57) 【要約】

【課題】 メモリリクエストがサービスを待機しなければならぬ時間を最短にすることにより帯域幅および並行性を最大にすべく、メモリスバシステムへのアクセスを制御する方法を提供することにある。

【解決手段】 マルチバンクインターリーブ形メモリシステム等のメモリシステムの並行性を最大にする順序で、メモリトランザクションを実行する。読取りデータが、読取りトランザクションがリクエストされた順序と同じ順序でバスに戻されるバッファメモリに収集される。適応アルゴリズムは、読取りからメモリへの書込みへの移行に関連するオーバーヘッドを最小にすべく書込みを集合化する。



【特許請求の範囲】

【請求項1】 データ処理システムにおける複数のメモリアクセスリクエスト再順序付けする方法において、複数のリクエストを受け入れる段階と、複数のリクエストから、利用可能なメモリ位置にアクセスするリクエストを選択する段階と、選択されたリクエストをスケジューリングする段階とを有することを特徴とする方法。

【請求項2】 前記リクエストを選択する段階が、利用可能なメモリ位置へのアクセスをリクエストする、複数のリクエストの中の最も古いリクエストを選択する段階からなることを特徴とする請求項1に記載の方法。

【請求項3】 前記リクエストを選択する段階が、複数のリクエストの中に、利用可能なメモリ位置にアクセスする読取りリクエストが存在するか否かを決定する段階と、該決定段階で、このような読取りリクエストが存在することを決定する場合には、利用可能なメモリ位置にアクセスする読取りリクエストを選択する段階と、前記決定段階で、利用可能な位置にアクセスする読取りリクエストが存在しないと決定する場合には、利用可能なメモリ位置にアクセスする非読取りリクエストを選択する段階とからなることを特徴とする請求項1に記載の方法。

【請求項4】 前記選択されたリクエストが書込みリクエストであるか否かを決定する段階と、選択されたリクエストが書込みリクエストであると決定された場合には、選択されたリクエストをスケジューリングする前記段階の後に、複数のリクエストから少なくとも1つの書込みリクエストを連続的にスケジューリングとを更に有することを特徴とする請求項3に記載の方法。

【請求項5】 複数のリクエストからの少なくとも1つのリクエストが戻り値を作り、複数のリクエストの戻り値を収集する段階と、複数のリクエストの元の順序付けに従って、収集された戻り値を順序付けする段階とを更に有することを特徴とする請求項1に記載の方法。

【請求項6】 前記複数のリクエストが、同じ同じメモリ位置への読取りリクエストおよび書込みリクエストを保有しているか否かを検出する段階を更に有することを特徴とする請求項1に記載の方法。

【請求項7】 前記各メモリ位置が複数のメモリバンクの1つと関連しており、前記リクエストを選択する段階が、複数のリクエストからのリクエストによりアクセスがリクエストされるメモリ位置を識別する段階と、識別されたメモリ位置が対応する複数のメモリバンクのうちの1つのメモリバンクを識別する段階と、識別されたメモリバンクが利用可能であるか否かを決定

する段階とからなることを特徴とする請求項1に記載の方法。

【請求項8】 メモリアクセスリクエストを再順序付けするデータ処理システムにおいて、複数のメモリアクセスリクエストを保持するリクエストバッファと、複数のメモリアクセスリクエストのうちの少なくとも幾つかのメモリアクセスリクエストにより探索されるメモリ位置の利用可能性を決定するための利用可能性決定器と、複数のメモリアクセスリクエストのうちから利用可能のあるメモリ位置にアクセスするメモリアクセスリクエストを選択すべく、前記利用可能性決定器に応答するリクエスト再順序付けユニットとを有することを特徴とするデータ処理システム。

【請求項9】 前記リクエスト再順序付けユニットが、利用可能のあるメモリ位置へのアクセスをリクエストする、複数のリクエストのうちの最も古いリクエストを選択することを特徴とする請求項8に記載のデータ処理システム。

【請求項10】 前記リクエスト再順序付けユニットが、利用可能なメモリ位置にアクセスする読取りリクエストが存在する場合に、該読取りリクエストを識別するための、前記利用可能性決定器に応答する読取り識別器と、利用可能なメモリ位置にアクセスする非読取りリクエストが存在する場合に、該非読取りリクエストを識別するための、前記利用可能性決定器に応答する非読取り識別器と、識別された読取りリクエストが存在する場合に該読取りリクエストを選択しかつ識別された読取りリクエストが存在しない場合に識別された非読取りリクエストを選択するコントローラとを有することを特徴とする請求項8に記載のデータ処理システム。

【請求項11】 前記選択されたリクエストが書込みリクエストである場合に、前記コントローラが少なくとも1つの非読取りリクエストを選択して複数のリクエストから利用可能なメモリ位置にアクセスすることを特徴とする請求項10に記載のデータ処理システム。

【請求項12】 前記複数のリクエストからの少なくとも1つのリクエストが戻り値を作り、複数のリクエストの元の順序付けに従って、複数のリクエストの戻り値を収集しかつ収集された戻り値を順序付けするデータ再順序付けユニットを更に有することを特徴とする請求項8に記載のデータ処理システム。

【請求項13】 前記リクエストバッファが、新しいメモリアクセスリクエストを複数のリクエスト内にダイナミックに受け入れることができることを特徴とする請求項8に記載のデータ処理システム。

【請求項14】 前記新しいリクエストが、前記複数の

リクエストのうちの書込みリクエストによりアクセスがリクエストされるメモリ位置と同じメモリ位置へのアクセスをリクエストするものであるか否かを検出する衝突検出器を更に有することを特徴とする請求項8に記載のデータ処理システム。

【請求項15】 前記複数のリクエストのうちの幾つかのリクエストにはデータが付随し、

選択されたメモリアクセスリクエストに付随する任意のデータを選択するリクエスト再順序付けユニットに応答するデータ再順序付けユニットを更に有し、選択リクエストおよび付随するデータが、メモリコントローラに受け入れられることを特徴とする請求項8に記載のデータ処理システム。

【請求項16】 各メモリ位置が、複数のメモリバンクのうちの1つのメモリ位置に関連しており、

利用可能性決定器が、メモリバンクが利用可能であるか否かを表示するメモリコントローラに接続されていることを特徴とする請求項8に記載のデータ処理システム。

【請求項17】 プログラムを実行しかつ複数のメモリアクセスリクエストのうちの少なくとも幾つかのメモリアクセスリクエストを発行する少なくとも1つのプロセッサと、

メモリ位置を備えたメモリシステムと、

該メモリシステムにより実行するための、選択されたメモリアクセスリクエストを受け入れるメモリコントローラとを更に有することを特徴とする請求項8に記載のデータ処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、メモリサブシステムを備えたデータ処理システムに関し、より詳しくは、帯域幅および並行性 (concurrency) を最大にし、これにより全メモリサブシステムおよびデータ処理システムの速度を増大させるべくメモリサブシステムへのリクエストを制御することに関する。

【0002】

【従来の技術】最近のデータ処理システムでは、メモリサブシステムの速度は、全システム速度に影響を与える主要な制限ファクタである。一般に、メモリアクセスは、コンピュータプロセッサおよびデータバスがメモリアクセスリクエストを発生しかつ搬送する速度より非常に遅いため、メモリボトルネックが存在する。メモリアクセスの速度が遅いことは、書込みリクエストではなく、読取りリクエストが存在するときに特に感じられる。これは、読取りリクエストは、リクエストプロセッサがデータを待機することを表示するためである。低メモリ速度により引き起こされるボトルネックは、コンピュータプロセッサの速度が、一般的なメモリコンポーネントの速度より高速で増大するときに、よりひどくなる。また、メモリボトルネックは、メモリサブシステム

を共有する多重プロセッサを有するコンピュータシステムおよびネットワークアーキテクチャが導入されるときに悪化する。

【0003】メモリボトルネックを緩和させる1つの慣用的なアプローチは、データ処理システム内の種々のレベルでデータキャッシングを使用することである。例えば、低速で安価なディスクメモリサブシステムのデータの一部は、高速システムRAM (ランダムアクセスメモリ) にコピーすなわち「キャッシュ (cached)」される。システムRAMのデータの一部は、次に、高速ではあるが高価な少数のRAMを保有する「第2レベル」キャッシュRAMサブシステムにキャッシュされる。データの一部も、同じチップにプロセッサとして存在する更に高速の「第1レベル」キャッシュメモリにキャッシュされる。データキャッシングは、低速メモリへのアクセスを最小にする有効な技術である。しかしながら、或る点で、種々のメモリレベルは依然としてアクセスする必要がある。従って、キャッシングを用いるか否かに係わらず、メモリアクセスをスピードアップする技術は依然として必要である。

【0004】メモリアクセスをスピードアップする試みとして、メモリを多重バンクに編成する方法がある。このメモリアーキテクチャーの下では、メモリの第1バンクとして繁忙サービスは第1バンクのメモリ位置にアクセスするリクエストであり、利用可能バンクは、次のリクエストが第2バンクのメモリ位置を目的とする場合に、次のメモリアクセスリクエストのサービスを開始する。メモリ位置は、連続的にアクセスされる連続メモリアドレスが異なるバンク内にあるように、バンク間でインターリーブされる。メモリバンクの慣用的使用に付随する問題は、アドレスがバンク間でインターリーブされる場合でも、連続アクセスリクエストが、依然として、時々、共通バンク内のアドレスを目的とすることである。この状況では、慣用的なメモリサブシステムは、メモリサブシステムが第2リクエストおよび任意の連続リクエストのサービスを開始する前に、共通バンクが利用できるようになることを依然として待機しなければならない。このような強制待機は無駄である。なぜならば、第3リクエストが異なる利用可能メモリバンクを目的とするため、さもなくば第3アクセスリクエストのサービスを開始できるからである。また、インターリーブされたバンク内へのメモリの単なる編成は、前述のように、読取りリクエストが対比書込みリクエストを有するという特別緊急性をアドレスしない。

【0005】

【発明が解決しようとする課題】当業界で要望されていることは、メモリアクセスがサービスを待機しなければならない時間を最短にすることにより帯域幅および並行性を最大にすべく、メモリサブシステムへのアクセスを制御する方法である。より詳しくは、先行リクエスト

が利用不可能メモリ位置を目的とするため未だサービスされない場合でも、メモリサブシステムが、利用可能メモリ位置にアクセスするリクエストのサービスを開始できるようにする方法が要望されている。また、特に、プロセッサが、次のタスクに進む前にメモリ書込みの完了を待機する必要がない「後書込み (posted-write)」システムでは、書込みリクエストより重要な読取りリクエストに特別な優先順位を与える方法が要望されている。

【0006】

【課題を解決するための手段】本発明は、メモリアクセスリクエストがサービスを行なうようにスケジュールされる順序を制御することによりメモリサブシステムの速度を増大させる方法および装置を提供する。本発明の一実施形態によれば、複数のメモリアクセスリクエストを再順序付けする方法が提供され、本発明の方法は、複数のリクエストを受け入れる段階と、複数のリクエストから、利用可能なメモリ位置にアクセスするリクエストを選択する段階と、選択されたリクエストをスケジューリングする段階とを有している。本発明の他の実施形態によれば、アクセスメモリへのリクエストを選択する段階が、複数のリクエストの中に、利用可能なメモリ位置にアクセスする読取りリクエストが存在するか否かを決定する段階と、このような読取りリクエストが存在することを決定する場合には、利用可能なメモリ位置にアクセスする読取りリクエストを選択する段階と、利用可能な位置にアクセスする読取りリクエストが存在しないと決定する場合には、利用可能なメモリ位置にアクセスする非読取りリクエストを選択する段階とからなる。

【0007】

【発明の実施の形態】本願明細書の以下の記載および図面を参照することにより、本発明の本質および長所が更に理解されよう。本発明は、データ処理システムにおいてリクエストされたメモリ演算がスケジュールされる順序を制御する技術に関する。図1は、本発明が具現されるコンピュータシステム101の簡略化されたブロック図である。この高レベルで示されたコンピュータシステムの形態は標準的なものであり、このため図1には「PRIOR ART」と記されている。しかしながら、このシステムがメモリへのアクセスを管理する本発明を採用しているものであれば、システム101のようなコンピュータシステムは従来技術ではない。既知のプラクティスによれば、コンピュータシステム101は、バスサブシステム111を介して多数の周辺装置と通信する1つ以上のプロセッサ103を有している。一般に、これらの周辺装置は、ランダムアクセスメモリ (RAM) サブシステム107およびディスクメモリサブシステム109等のメモリサブシステムと、キーボード104またはマウス105等の入力装置と、ディスプレイ102のような出力装置とを有している。他の一般的な周辺装置 (図示せず) として、プリンタ、テープメモリサブシステム、遠

隔ネットワーク形サーバメモリサブシステム等がある。

【0008】本願明細書では、用語「バスサブシステム」は、システム101の種々のコンポーネントを、意図するままに相互通信させるあらゆる機構を包含するものとして広範囲に使用される。例えば、バスサブシステム111は単一バスとして概念的に示されているが、一般的なコンピュータシステムは、ローカルバス、1つ以上の拡大バス、直列ポート、並列ポート、ネットワーク接続部等の多数の「バス」を有している。一般に、コンピュータシステム101のコンポーネントを、同じ物理的配置にする必要はない。図1は、本発明を具現するコンピュータシステムを示すものであるが、単なる一形式である。当業者ならば、多くのコンピュータシステム形式および構成が本発明の具現に適していることが容易に理解されよう。

【0009】図2は、本発明の多重プロセッサの実施形態201を示すブロック図である。データ処理システム201は、多数のプロセッサP1~P4 (それぞれ、参照番号203、204、205、206で示されている) を有している。これらのプロセッサは、メモリバス215に接続されている。メモリバス215は、リクエスト再順序付けユニット214を介して、メモリサブシステム216に接続されている。プロセッサP1~P4は、メモリサブシステム216に対するリクエストをメモリバス215に発生する。リクエスト再順序付けユニット214は、メモリバス215からのリクエストを受け入れ、該リクエストを、後述するようにして、本発明の技術に従ってスケジュールする。或る実施形態では、メモリバス215はIntel P6バス、プロセッサの個数は4つ以下、プロセッサはIntel PentiumPro-compatible プロセッサである。

【0010】図3は、二重ブリッジ形メモリバスを備えた本発明の多重プロセッサの実施形態を示すブロック図である。データ処理システム202は、多数のプロセッサP1~P8 (それぞれ、参照番号223、224、...、230で示されている) を有している。これらのプロセッサのうち、P1~P4は第1メモリバス211に接続され、プロセッサP5~P8は第2メモリバス212に接続されている。2つのメモリバス211、212は、これらの両バスおよび第3バス215の間の切換えを遂行するコントローラ/クロスバースイッチ213により架橋されている。第3バス215は、リクエスト再順序付けユニット214を介して、メモリサブシステム216に接続されている。バス211、212、215は、バスサブシステムに属するものと考えられる。

【0011】コントローラ/クロスバースイッチ213は、干渉性チェックを遂行しかつバス211、212のうちの一方のバスから他方のバスへの演算を反映するのに必要な情報をルーチングする。プロセッサP1~P4は、メモリサブシステム216に対するリクエストを第

1メモリバス211に発生する。プロセッサP5~P8は、メモリサブシステム216に対するリクエストを第2メモリバス212に発生する。コントローラ/クロスバススイッチ213は、2つのメモリバス211、212からのメモリリクエストを第3バス215にルーティングする。リクエスト再順序付けユニット214は、第3バス215からのリクエストを受け入れ、該リクエストを、後述するようにして、本発明の技術に従ってスケジューリングする。或る実施形態では、各メモリバス211、212はIntel P6バス、各メモリに接続されるプロセッサの個数は4つ以下、プロセッサはIntel Pentium Pro-compatibleプロセッサである。

【0012】図4は、発明の背景の部分で論じた従来技術によるリクエストされたメモリ演算の順序付けを示す概略図である。図示を簡単化することのみの理由で、図示の全てのリクエストされたメモリ演算は読取り演算である。図4において、複数のリクエストされたメモリ演算303（各リクエストされたメモリ演算は目的メモリアドレスを有している）は、データ処理システムの一部301によりメモリサブシステム216に提供される。リクエスト/メモリアドレスは、A110、A104、A99、A50、A2、A1（それぞれ、参照番号305、306、・・・310で示されている）の順序をなしている。メモリサブシステム216は、各リクエストされた演算の遂行を順に開始する。リクエストされた演算が利用可能である場合には、メモリサブシステム216はリクエストされた演算の遂行を開始する。そうでない場合には、メモリサブシステムは目的メモリアドレスが利用可能になることを待機する。この待機は、発明の背景の部分で説明した、利用可能アドレスを目的とする連続リクエストされた演算がサービスされ始める状況では無駄なことである。

【0013】リクエストされた演算303によりリクエストされたデータは、該リクエストされた演算303の順序に一致する順序312で戻される（322）。この例では、データの順序312は、D111、D104、D99、D50、D2、D1（それぞれ、参照番号315、316、・・・320で示されている）である。図5は、本発明のリクエストされたメモリ演算の再順序付けを広く示す概略図である。図示を簡単化することのみの理由で、図示の全てのリクエストされた演算は読取り演算である。図5において、複数のリクエストされた演算303（各リクエストされたメモリ演算は目的メモリアドレスを有している）は、データ処理システムの一部301により提供される。リクエスト/メモリアドレス303は、A110、A104、A99、A50、A2、A1（それぞれ、参照番号305、306、・・・310で示されている）の初期順序をなしている。これらのリクエストされた演算は、連続的遂行を行なう新しい順序付け311に従ってメモリサブシステム216に

提供される（323）。再順序付けは、利用不可能な目的アドレスにより引き起こされるメモリサブシステム216での待機を最短にするように、後述の技術に従って遂行される。リクエスト/アドレスの順序は、この例では、A99、A110、A1、A104、A2、A50の順序に最適化される（311）。

【0014】データは、D99、D110、D1、D104、D2、D50の最適順序で、メモリサブシステム216から読み取られる（324）。これらのデータは、次に、D110、D104、D99、D50、D2、D1（それぞれ、参照番号315、316、・・・320で示されている）の順序312に再順序付けされる（313）。再順序付けされたデータ順序は、初期リクエスト順序と一致する。再順序付けされたデータは、戻される（322）。データは、初期リクエスト順序に従って戻されるので、例えばプロセッサのようなメモリ演算をリクエストする構成要素（図示せず）は、リクエストされた演算がメモリサブシステム216により実際に遂行される順序を意識する必要はない。図5には、図示を容易化する目的で、リクエストされた読取り演算のみが示されている。一般に、リクエストされた演算も書込み演算である。書込みのためのデータは、書込みリクエスト/アドレス自体と一緒に再順序付けされなくてはならない。一般に、書込みの前に読取りが遂行されるように、同じアドレスへのリクエストされた読取りが続く、リクエストされた書込みが再順序付けされる状況を防止するステップをとらなくてはならない。不正確な読取りを防止する1つの方法は、書込みリクエストが既にスケジュールされたことをシステムの一部301が演繹するまで、該部分301が、読取りリクエストが続く書込みリクエストを発行しないようにすることである。他の方法は、システムの一部301が、制限されることなくリクエストを発行できるようにし、次に、再順序付けプロセス自体の間に前記状況を積極的に防止することである。後者の方法では、例えばプロセッサのようなメモリ演算をリクエストする構成要素（図示せず）は、再順序付けが行なわれることを意識する必要はない。

【0015】図6は、本発明を具現するデータ処理システム401を示す機能的ブロック図である。システムの一部301は、メモリアクセスリクエスト303およびリクエスト303の或るものと関連する「W」データ403を発行する。「W」データは、リクエストされたメモリ演算（単一および複数）と関連するデータであり、目的メモリ位置（単一および複数）に書き込まれるべきデータを有している。リクエスト303および「W」データ403は、例えば、データ処理システムの一部301におけるバスサブシステム（図示せず。このバスサブシステムは図1のバスサブシステム111と同じである）から発行される。リクエスト再順序付けユニット214内で、アドレス再順序付けサブユニット311がリ

クエスト303を受けかつこれを一時的にバッファに記憶する。本発明の実施形態では、アドレス再順序付けサブユニット311は、オプションとしての衝突検出器404を介して、クエスト303を受ける。アドレス再順序付けサブユニット311に入るとき、クエスト303には初期順序を付すことができる。アドレス再順序付けサブユニット311は、下記の技術に従って、クエストの新しい順序408で、メモリサブシステム216にクエストを提供する。

【0016】クエスト再順序付けユニット214内の第1データ再順序付けサブユニット406は、「W」データ403を受ける。第1データ再順序付けサブユニット406は、アドレス再順序付けサブユニット311からの方向で、「W」データの新しい順序410で、「W」データをメモリサブシステム216に提供する。「W」データ403の新しい順序410は、クエスト303の新しい順序408に一致する。メモリサブシステム216はクエストを満たしかつ得られた全ての「R」データ412をクエスト再順序付けユニット214に戻す。「R」データはクエストされたメモリ演算（単一または複数）から得られ、目的メモリ位置（単一または複数）からのデータ読取りを含んでいる。

【0017】クエスト再順序付けユニット214内で、第2データ再順序付けサブユニット313が「R」データを受けかつ該「R」データをデータ処理システムの一部301に戻す。クエスト303が初期順序を有する場合には、第2データ再順序付けサブユニット313は、「R」データを戻す（414）前に、「R」データ412への順序を復元する（これにより、復元された順序はクエスト303の初期順序に一致する）。本発明の実施形態では、第2データ再順序付けサブユニット313および第1データ再順序付けサブユニット406は、アドレス再順序付けサブユニット311からの方向を受けるべく接続された単一ユニットにおいて実施される。図6の本発明の実施形態は、一般に、進行中の実行時ダイナミック態様で演算する。クエスト再順序付けユニット214は、新しいクエストおよび対応「W」データをダイナミックに受け入れる。クエスト再順序付けユニット214は、対応する「W」データのクエストおよび410のための新しい最適順序408に従って、クエストおよび対応する「W」データをメモリサブシステム216にダイナミックに提供する。クエスト再順序付けユニット214は、メモリサブシステム216からの「R」データを、ダイナミックに受け、再順序付けしかつ戻す。

【0018】本発明の幾つかの実施形態の衝突検出器404は、当該アドレスへの早期のあらゆる書込みが完了するまで、アドレスからの読取りが起らないように確保する1つの方法を実施する。このように確保することにより、衝突検出器404は、クエスト再順序付けに

よって、アドレスからの早期で誤差のある読取りを行なうという上記問題を引き起こすことを防止する。衝突検出器404は次の段落で説明するように作動し、上記問題を引き起こすリクエストシーケンスが、第1位置でアドレス再順序付けサブユニット311に入ることを防止する。早期読取りの問題を防止する他の方法は、本発明の技術に基づくものであることは明白であろう。例えば、衝突検出器404をもたない本発明の実施形態では、アドレス再順序付けサブユニット311自体がリクエストシーケンスをモニタリングし、かつ同じアドレスへの読取りリクエストが続く書込みリクエストの順序を反転させないようにする。

【0019】衝突検出器404では、入ってくる読取りリクエストの目的アドレスが、アドレス再順序付けサブユニット311に待機する全てのリクエストのアドレスに対してチェックされる。読取りリクエストが書込みリクエストのアドレスの1つと一致する場合には、読取りリクエストが停止され、読取りリクエストがアドレス再順序付けサブユニット311に入ることを防止する。一致する書込みリクエストが最終的にアドレス再順序付けサブユニット311を出てメモリサブシステム216に向かうと、「衝突」が消滅しかつ読取りリクエストがアドレス再順序付けサブユニット311に入ることが可能になる。クエスト再順序付けユニット214に入ると、クエストは、クエストの初期順序を定める省略時優先順位（default priorities）を有する。クエストの新しい順序を確立することにより、本発明は、クエストに新しい優先順位を割り当てる。現在利用可能なアドレスを目的とするこれらのクエストは、高い優先順位で割り当てられる。

【0020】クエストの省略時優先順位は、任意の基準すなわち発見的規則（heuristic rules）により決定される。本発明の実施形態では、新しいクエストより高い優先順位をもつものとして、古いクエストが定められる。クエストのエージ（age）は、例えば、クエストが発行された時、クエストが受けられた時、またはクエストに関するタイムスタンプにより決定される。本発明の他の実施形態では、省略時優先順位は、クエストに関連する優先順位フィールドにより決定される。省略時優先順位はまた、最初にクエストを発した構成要素（例えば、プロセッサ）のアイデンティティによっても影響を受ける。本発明の実施形態では、クエスト再順序付けユニット214は、プロセッサで遂行されるようにソフトウェアに実施される。本発明の実施形態では、ソフトウェアは、コンピュータが読取り可能な記憶媒体を備えたコンピュータプログラム製品に記憶される。コンピュータが読取り可能な記憶媒体として、リードオンリメモリ（ROM）、プログラマブルリードオンリメモリ（PROM）、CD-ROM、ハードディスク等がある。本発明の実施形態では、クエスト再順序付

けユニット214はソフトウェアで構成できる。

【0021】図7は、本発明の実施形態によるアドレス再順序付けサブユニット311でのリクエスト順序付けを示すフローチャートである。図6を参照しつつ、図7を説明する。図7の実施形態では、アドレス再順序付けサブユニット311は、ステップ502において、再順序付けサブユニット311でバッファされたリクエストが、利用可能なアドレスを目的としているか否かを決定する。YES(503)であれば、ステップ504において、再順序付けサブユニット311が、新しい順序408へのリクエストをスケジュールし、従ってデータ再順序付けサブユニット406に対応するあらゆる「W」データをスケジュールさせる。スケジュールされたリクエストは、ステップ504で、アドレス再順序付けサブユニット311から取り出される。本発明の好ましい実施形態では、アドレス再順序付けサブユニットは、ステップ504で、利用可能なアドレスを目的とするリクエスト間の最高の省略時優先順位を有するリクエストをスケジュールする。

【0022】リクエストのスケジューリングは、リクエストがメモリサブシステム216に供給するリクエストのキューに入れられる。本発明の好ましい実施形態では、キューの長さはゼロであり、リクエストはこれらがスケジュールされるやいなや、メモリサブシステム216に提供される。本発明の実施形態では、メモリサブシステムはメモリのバンクを有し、メモリアドレスは、メモリアドレスが存在するバンクが利用可能であるときに利用可能である。本発明の実施形態では、メモリサブシステムは、インターリーブされたアドレスを備えたメモリのバンクを有し、メモリアドレスは、メモリアドレスが存在するバンクが利用可能であるときに利用可能である。本発明は、メモリのバンクを備えたメモリサブシステムへのアクセスを制御するのに特に適している。また、本発明は、インターリーブされたアドレスを備えたメモリのバンクを備えたメモリサブシステムへのアクセスを制御するのに特に適している。また、本発明は、半導体メモリサブシステムまたは同様なメモリサブシステムまたはより高速のメモリサブシステムのようなメモリサブシステムへのアクセスを制御するのに特に適している。

【0023】図8は、書込みリクエストに関して読取りリクエストを好都合にする本発明の実施形態のリクエストスケジューリングを示すフローチャートである。発明の背景のセクションで述べたように、読取りリクエストは、リクエストングプロセッサがデータを待機することを表示するので、できる限り迅速に読取りリクエストを取り扱うことが特に有効である。従って、読取りリクエストは、他の演算(例えば書込み演算)よりも厳格なリクエスト形式である。図6を参照しつつ、図8を説明する。図8の実施形態では、アドレス再順序付けサブユ

ニット311は、まずステップ610で、再順序付けサブユニット311でバッファされるリクエストの読取りリクエストが利用可能なアドレスを目的とするか否かを決定する。YESであれば(611)、アドレス再順序付けサブユニット311は、ステップ612で、新しい順序付け408への読取りリクエストをスケジュールする。本発明の好ましい実施形態では、アドレス再順序付けサブユニットは、ステップ612は、利用可能なアドレスを目的とする読取りリクエスト間の最高の省略時優先順位を有するリクエストをスケジュールする。スケジュールされた読取りリクエストは、ステップ612で、再順序付けサブユニット311から取り出される。

【0024】ステップ610において、再順序付けサブユニット311の読取りリクエストが利用可能なアドレスを目的としないことを決定すると(613)、再順序付けサブユニット311は、ステップ614において、再順序付けサブユニット311のリクエストのうちの書込みリクエストが利用可能なアドレスを目的とするか否かを決定する。YESであれば(616)、アドレス再順序付けサブユニット311は、ステップ618において、新しい順序付け408への書込みリクエストをスケジュールしかつデータ再順序付けサブユニット406に任意の対応する「W」データをスケジュールさせる(410)。本発明の好ましい実施形態では、ステップ618において、利用可能なアドレスを目的とする書込みリクエスト間の最高の省略時優先順位を有するリクエストをスケジュールする。スケジュールされた書込みリクエストは、ステップ618で、再順序付けサブユニット311から取り出される。

【0025】多くのメモリサブシステムでは、書込みリクエストは、メモリサブシステムに入る書込みデータとメモリサブシステムを出る読取りデータとの間のメモリサブシステムのポテンシャルバスコンテンションのため、読取りリクエストがスケジュールされた直後にスケジュールされることはない。本発明がこのようなメモリサブシステムへのアクセスを制御するのに使用されるとき、この可能性のあるコンテンションを考慮すべきである。可能性のあるコンテンションを防止する1つの方法は、利用可能なアドレスを目的とする書込みリクエストが存在するか否かだけでなく、書込みデータがクリアされているか否かを尋ねるようにステップ614を変更することである。例えば、前にスケジュールされたリクエストが読取りリクエストでない場合、または前にスケジュールされたリクエストの実行時間と問題の書込みリクエストの実行時間との間で確保されている場合には、書込みバスはクリアされていると決定できる。書込みリクエストが存在しかつ書込みバスがクリアされている場合にのみ(616)、書込みリクエストがステップ618でスケジュールされる。

【0026】前の段落で説明したメモリサブシステムで

は、バスターンアラウンド中または読取りリクエストの取扱いから書込みリクエストの取扱いへの移行中に時間が喪失される。喪失時間を短縮させるため、本発明の実施形態は、オプションとして、ステップ618の後に、ターンアラウンドを最小にするバースト書込みステップを設けることができる。バースト書込みステップ620は、ステップ618で書込みリクエストがスケジュールされたならば、書込みリクエストのバーストがステップ620で連続的にスケジュールされる挙動を実施する。この場合、書込みリクエストは、グループとして一体化されるのが有効なため、「バースト可能」な形式である。図9は、図8のバースト書込みステップを示すフローチャートである。図9は、殆ど説明を要しないものである。図9から分かるように、利用可能なアドレスを目的とする書込みリクエストがもはや存在しなくなるか(626)、所定数Xの書込みリクエストが現在のバースト中にスケジュールされかつ利用可能なアドレスを目的とする読取りリクエストが存在するようになる(626)まで、ステップ622で、利用可能なアドレスへの書込みリクエストがスケジュールされる。本発明の一実施形態では、個数Xはシステム条件に基づいて調節される。例えば、Xの値は、ソフトウェアプログラム、ディップスイッチ等によりレジスタに記憶することができる。図3に示すような8個のIntel Pentium Pro プロセッサを備えた本発明の実施形態では、Xの値として4が適していることが判明している。

【0027】図10は、インターリーブされた多重バンクメモリシステムへのリクエストを制御する本発明の実施形態701を示すブロック図である。図10において、バス111は、リクエストバッファ703と呼ばれるシフトバッファの入力エンド705に、メモリリクエストを送る。リクエストバッファ703は、要素705のような多数の要素(各要素は1つのリクエストを記憶できる)を有している。8個の要素をもつバッファは、本発明の8個のIntel Pentium Pro プロセッサの実施形態に適していることが判明している。要素内には、「形式」707、「ID」709および「バンク」711としてリクエストが記憶される。リクエストの形式707は、「読取り」または「書込み」の値をとることができる。リクエストのID709は、リクエストの目的アドレスを特定化する。本発明の一実施形態では、リクエストのID自体はアドレスではなく、インデックス、すなわちアドレスを保持するアドレスバッファ(図示せず)へのポインタである。リクエストのバンク711は、目的アドレスが所属するメモリのバンクである。各要素はまた、要素が現在リクエストを保有しているか否かを告げる「有効」ビット713を保有している。

【0028】リクエストバッファ703は、読取り717または書込み719にメモリのどのバンクが現在利用可能であるかを表示するメモリコントローラ715から

の情報を受けるように接続されている。どのバンクが利用可能であるかに関する情報に基づいて、リクエストバッファは、どのリクエストが利用可能アドレスを目的としているかを決定する。リクエストバッファ703は、利用可能なアドレスを書込みMUX727に目的付ける全ての書込みリクエストを与える(725)。読取りMAX723は、読取り/書込みスケジューラ731への目的利用可能な読取りリクエスト721の最も古い読取りリクエストを与える。また、読取りMAX723は、最も古い目的利用可能な読取りリクエスト729の存在を表示する「利用可能な読取り」フラグ735を発生する。読取り/書込みスケジューラ731および優先ジェネレータ737は、「利用可能な書込み」フラグ739を受ける。

【0029】読取り/書込みスケジューラ731および優先ジェネレータ737は、協働して実施形態701の再順序付け規則(reordering rules)を実施する。実施形態701の再順序付け規則について、図8および図9に関連して説明した通りである。優先ジェネレータ737は、後述のように、読取りリクエストの現在の優先順位741および書込みリクエストの現在の優先順位743を決定する。一部が現在の優先順位に基づいて、読取り/書込みスケジューラ731は、後述のように、最も古い目的利用可能な読取りリクエスト729または最も古い目的利用可能な書込みリクエスト733をスケジュールするか(745)。ひとたびリクエストがスケジュールされると(745)、メモリコントローラ715はリクエストを受け入れ、優先ジェネレータ737およびシフトコントローラ751は、読取りリクエスト747または書込みリクエスト749には、スケジュールされたか否かが知らされる。

【0030】シフトコントローラ751はリクエストバッファ703のコンテンツを知るため接続される。リクエストがスケジュールされた後、シフトコントローラ751は、リクエストバッファ703からの当該リクエストを取り出す。これを行なうため、シフトコントローラ751は2つの機能を遂行する。第1の機能は、シフトコントローラ751が、今スケジュールされたリクエストの形式(すなわち、読取り形式747または書込み形式749)に一致する最も古い目的利用可能なリクエストを識別することにより、どのリクエストが今スケジュールされたものであるかを演繹することである。第2の機能は、シフトコントローラ751が、リクエストバッファに情報を提供して、1つの読取りリクエストにより今スケジュールされたリクエストより古い全てのリクエストを、リクエストバッファ703の入力エンド705(すなわち、新しいエンド)からシフトすることである。

【0031】上記のように、リクエストバッファ703は、若い方のエンド705および反対側の古い方のエン

ド706を備えたシフトレジスタとして構成されている。バッファ703にリクエストが入るエイジは、必然的に、若い方のエンド705から古い方のエンド706に向かって増大するリクエストバッファ703内の流れは、全ての要素に同時に加えられるバッファの各要素Xについての下記規則を用いて制御される。これらの規則の効果は、リクエストバッファ703がそのコンテンツをシフトして非占拠要素を満たし、これにより、若い方のエンド705での充填のための若い方の要素の利用可能性を最高にする。

規則1： 要素Xまたは古い要素が使用されない場合には、要素Xのコンテンツが隣接する若い要素に置換される。

【0032】規則2： 要素Xまたは古い要素がメモリシステムに行くことをスケジュールされている場合には、要素Xのコンテンツが隣接する若い要素に置換される。

規則3： 上記両規則1および2の条件が満たされない場合には、要素Xのコンテンツが不変に維持される。

実施形態701は衝突検出器404（図10には示されていない）を用いて作動するように設計されているため、アドレスへの能動書込みリクエストには同じアドレスへの能動リクエストが続かないものと仮定する。実施形態701は、衝突検出器をリクエストバッファ703に付加することにより、このような仮定を行なわないように変更できる。図11は、読取り741のための現在の優先順位を与える、図10の読取り／書込みスケジューラ731のリクエストスケジューリングを示すフローチャートである。図11において、リクエストが利用可能なアドレスを目的とする場合には（803）、最も古い読取りリクエストがスケジュールされる（801）。そうでない場合には（805）、前の読取りリクエストが書込みバスをクリアするようにスケジュールされてから十分な時間が経過している場合には、利用可能なアドレス809を目的とする最も古い書込みリクエストがスケジュールされる（807）。

【0033】図12は、書込み743のための現在の優先順位を与える、図10の読取り／書込みスケジューラ731のリクエストスケジューリングを示すフローチャートである。図12において、リクエストが利用可能なアドレスを目的とする場合には（823）、最も古い読取りリクエストがスケジュールされる（821）。そうでない場合には（825）、利用可能なアドレス829を目的とする最も古い読取りリクエストがスケジュールされる（827）。図13、図14、図15は、図10の読取り／書込み優先ジェネレータ737での読取り／書込み優先ジェネレーションを示すフローチャートである。図13から分かるように、目的利用可能な読取りリクエストが存在しない場合（905）および目的利用可能な書込みリクエストが存在する場合（909）にの

み、読取り741の現在の優先順位が書込みの優先順位に変更される（907、913）。必要ならば（910、912）、書込みの現在の優先順位を確立する前に、優先ジェネレータが待機する（915）。この待機は、前の読取りリクエストがスケジュールされた後に十分な時間経過を与えることにより（911）、クリアな書込みバスを確保するためのものである。待機の間、書込み優先順位および読取り優先順位のいずれも主張されない（すなわち、優先順位は決定されない）。

【0034】図15から分かるように、現在の優先順位が主張されないと、この状態は、目的利用可能な書込みリクエストが存在せず（919）かつ所定数Xの書込みが連続的に書き込まれなくなるまで維持される（917）。本発明の8個のPentiumProを用いた実施形態において首尾よく作動するための、Xについての4つの値が見出されている。以上、本発明の特定実施形態について全て説明したが、種々の変更を行なうことができる。従って、上記説明は、本発明の範囲を制限するものではなく、本発明およびその均等物の範囲は特許請求の範囲の記載により定められる。

ソースコード付録

マイクロフィッシュソースコード付録として、本発明の一実施形態を実施するソースコードがある。このソースコードは、回路を記載する産業において使用されているVerilogのハードウェア記載言語にある。適当なコンパイラでコンパイルするとき、ソースコードは本発明のシミュレーションを行なうことができる。適当なコンパイラとして、例えばSun Microsystems社のSparcワークステーションに使用できる、Cadence Design Systems社から市販されているTurbo-Verilogがある。ソースコードに使用されるVerilogのサブセットは、Synopsys, Inc.社から市販されている標準合成ツールを用いてゲートへの合成、従って集積回路（IC）を選択する実施形態が可能である。

【図面の簡単な説明】

【図1】本発明が具現されるコンピュータシステムを示すブロック図である。

【図2】本発明の多重プロセッサの実施形態を示す高レベルブロック図である。

【図3】二重ブリッジ形メモリバスを備えた本発明の多重プロセッサの実施形態を示す高レベルブロック図である。

【図4】従来技術によるリクエストされたメモリ演算の順序を示す概略図であり、リクエストされた読取り演算を示すものである。

【図5】本発明によるリクエストされたメモリ演算の順序を示す概略図であり、リクエストされた読取り演算を示すものである。

【図6】本発明の一実施形態を示すブロック図である。

【図7】本発明の実施形態におけるリクエスト順序を示

すフローチャートである。

【図8】本発明の読取りに適した実施形態におけるリクエストスケジューリングを示すフローチャートである。

【図9】本発明の読取りに適した実施形態におけるリクエストスケジューリングを示すフローチャートである。

【図10】本発明の一実施形態を示す回路ブロック図である。

【図11】図10の読取り／書込みスケジューラのリクエストスケジューリングを示すフローチャートである。

【図12】図10の読取り／書込みスケジューラのリクエストスケジューリングを示すフローチャートである。

【図13】図10の優先ジェネレータにおける読取り／書込み優先ジェネレーションを示すフローチャートである。

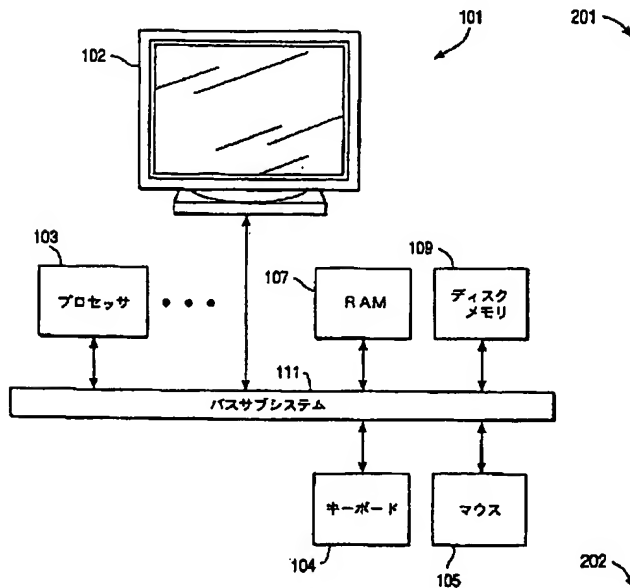
【図14】図10の優先ジェネレータにおける読取り／書込み優先ジェネレーションを示すフローチャートである。

【図15】図10の優先ジェネレータにおける読取り／書込み優先ジェネレーションを示すフローチャートである。

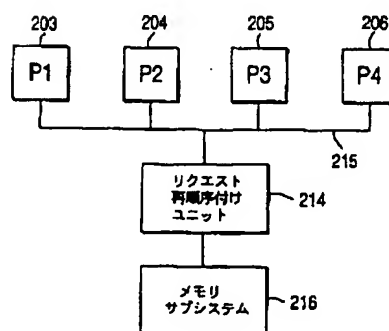
【符号の説明】

- 101 コンピュータシステム
- 102 ディスプレイ
- 103 プロセッサ
- 104 キーボード
- 105 マウス
- 107 ランダムアクセスメモリ (RAM)
- 109 ディスクメモリサブシステム

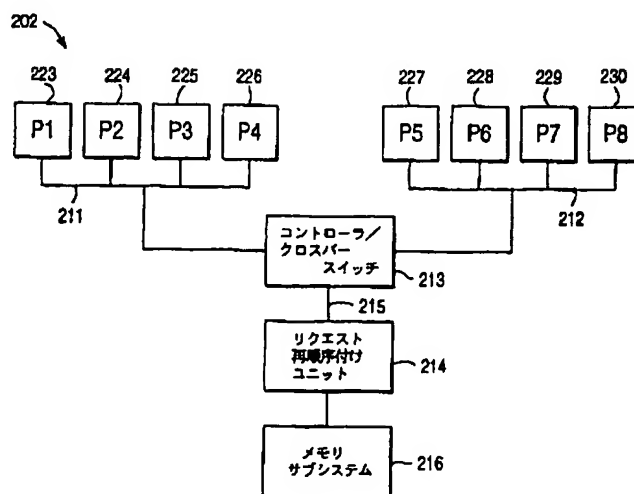
【図1】



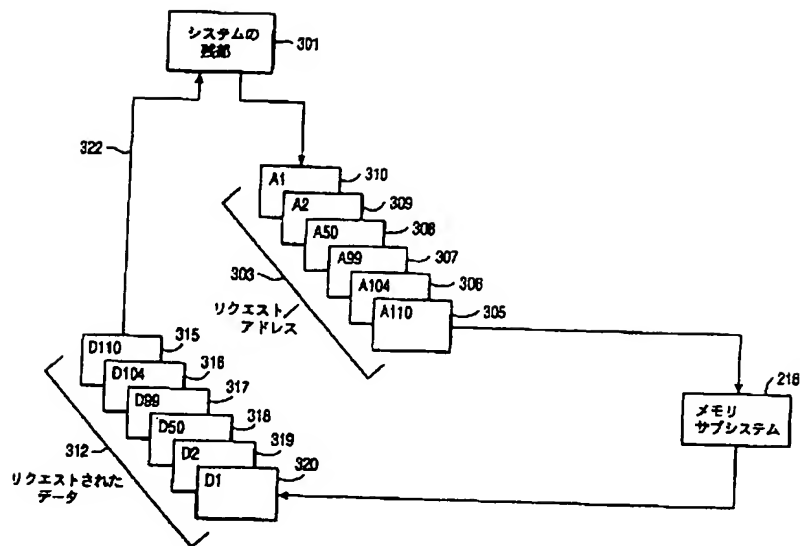
【図2】



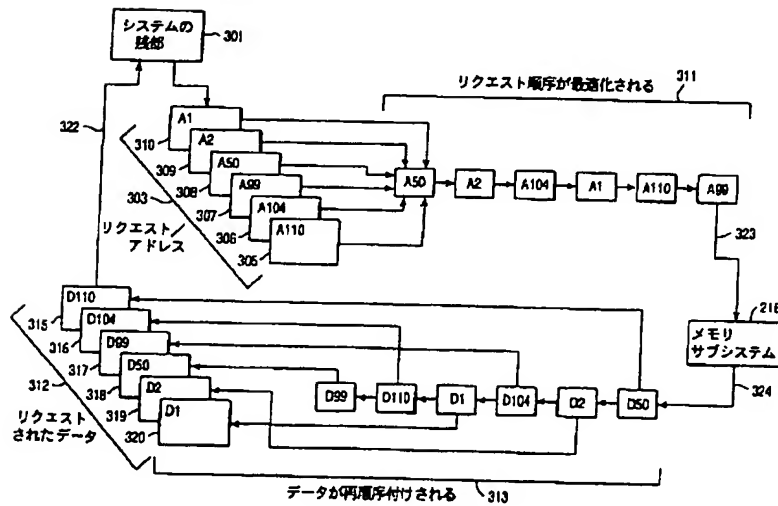
【図3】



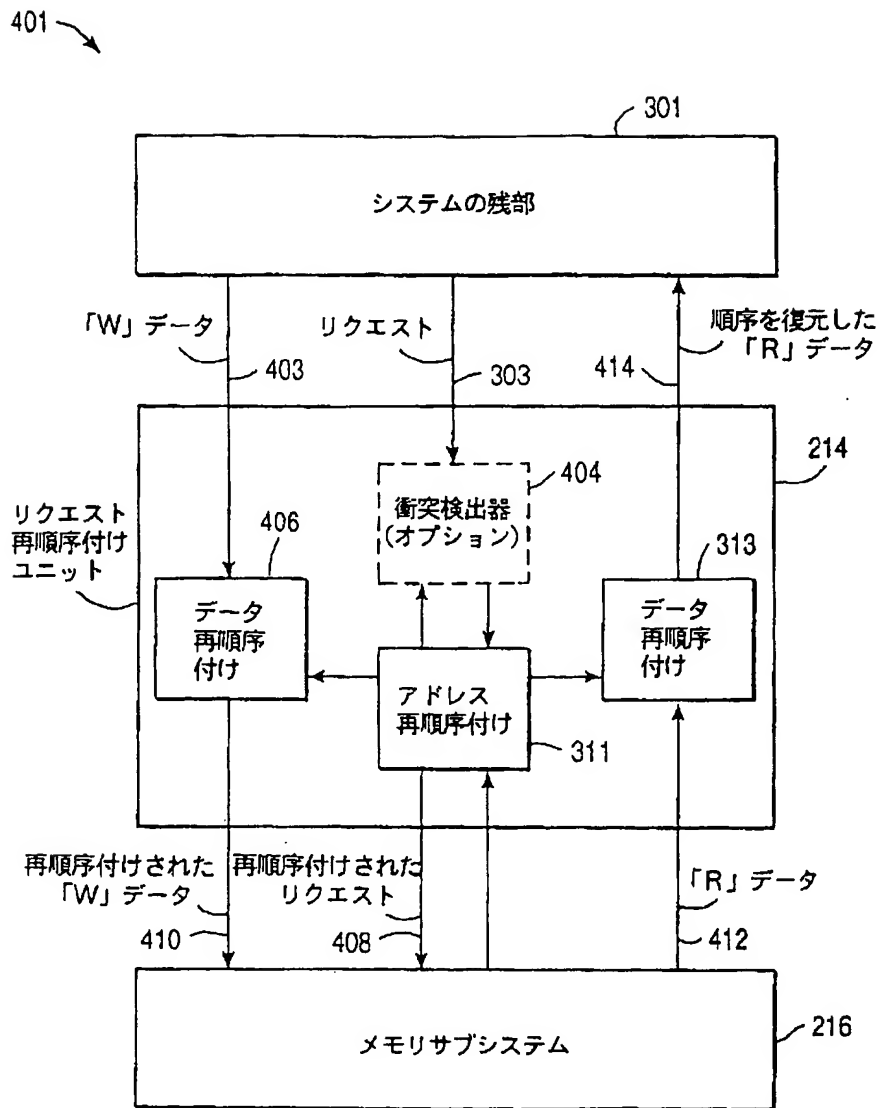
【図4】



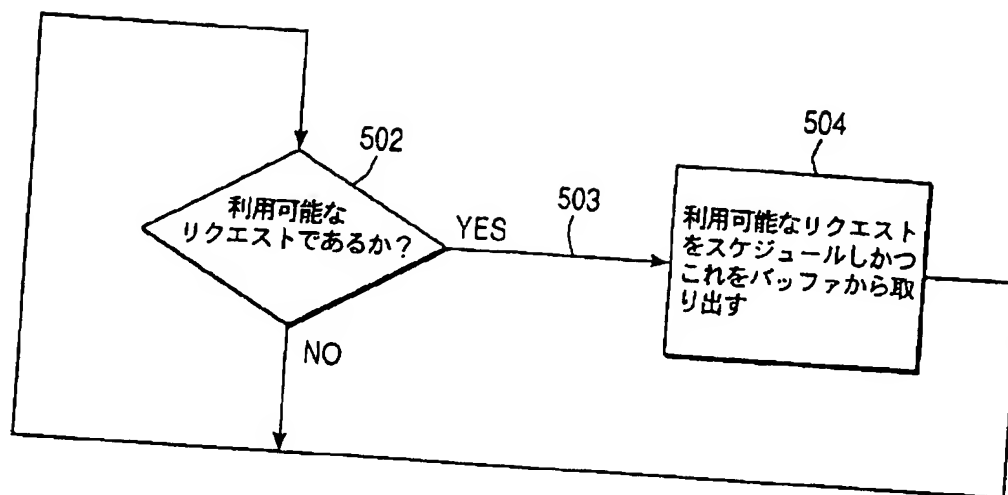
【図5】



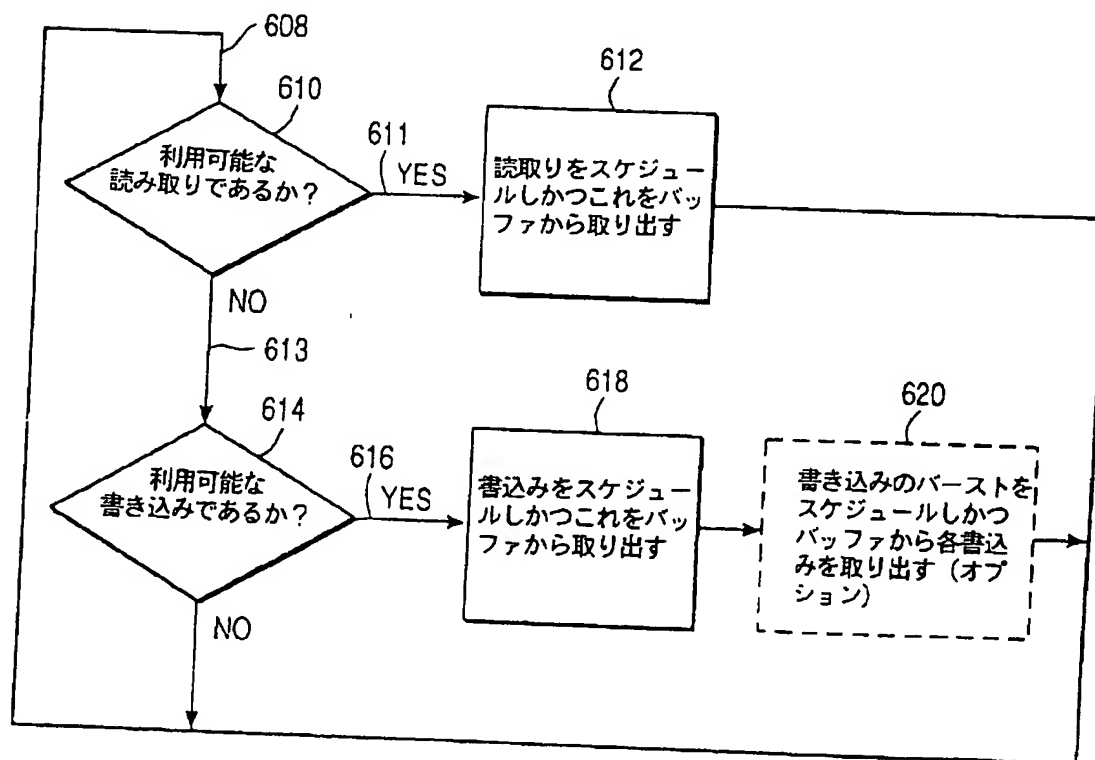
【図6】



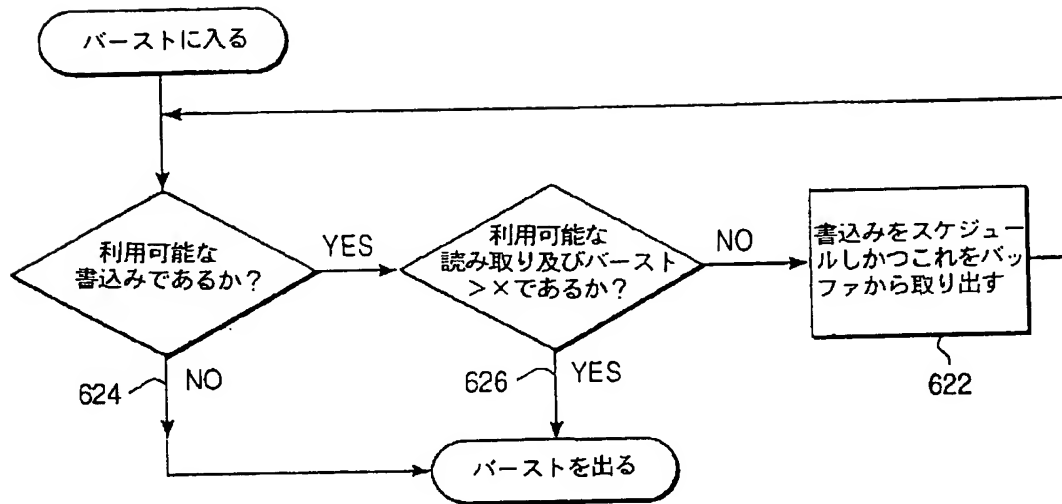
【図7】



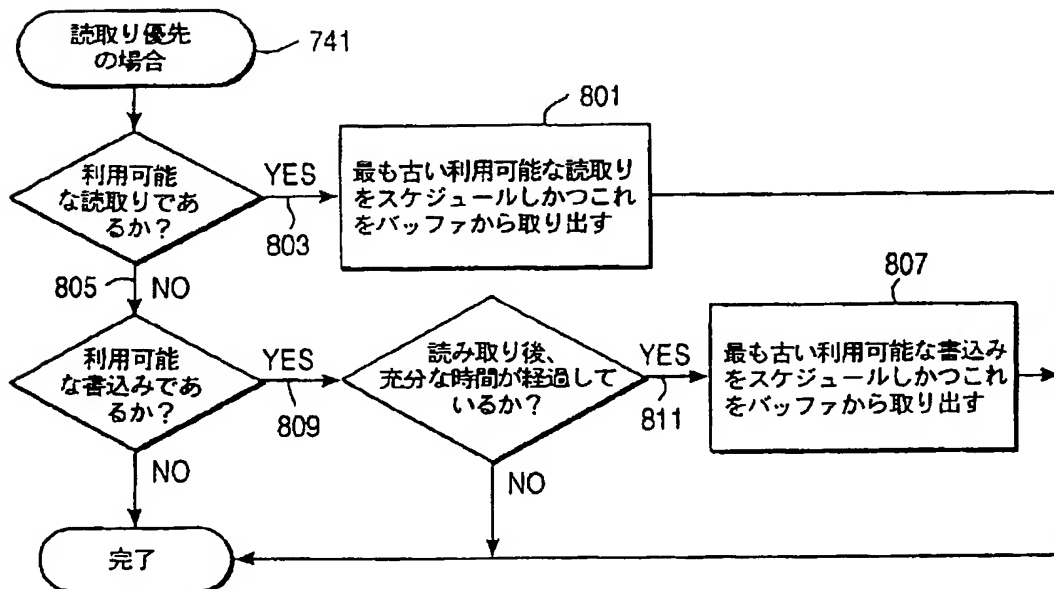
【図8】



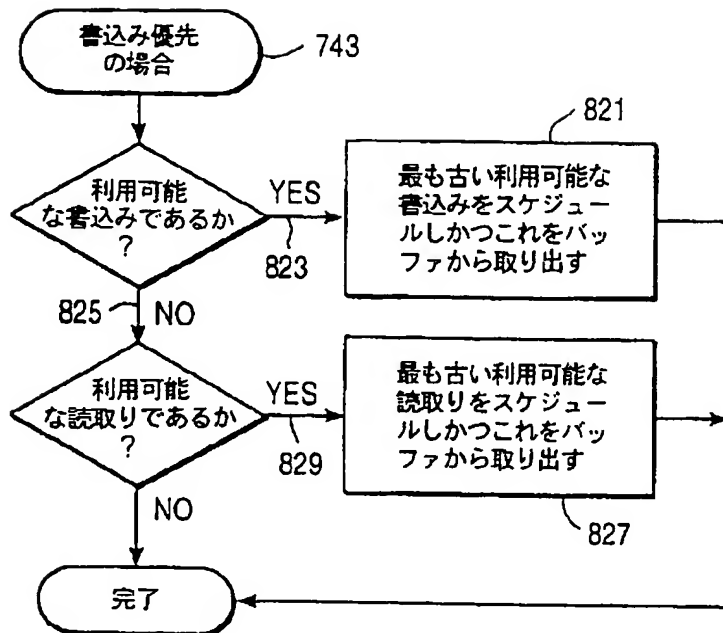
【図9】



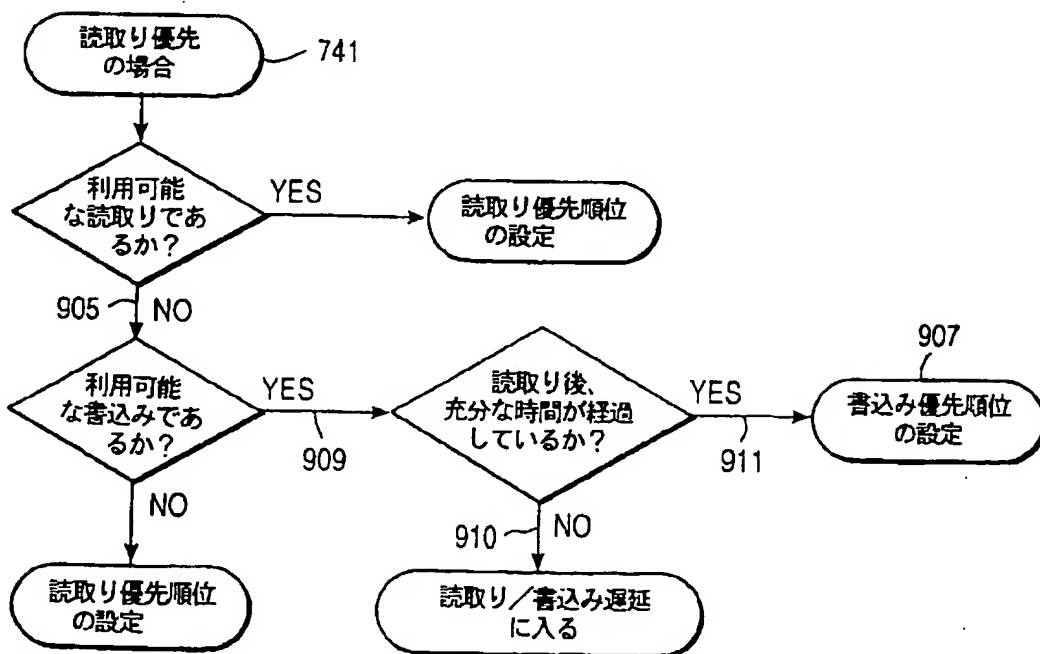
【図11】



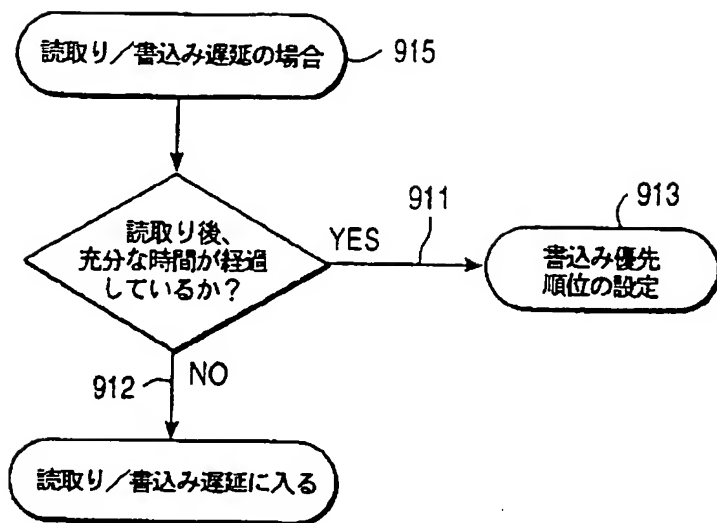
【図12】



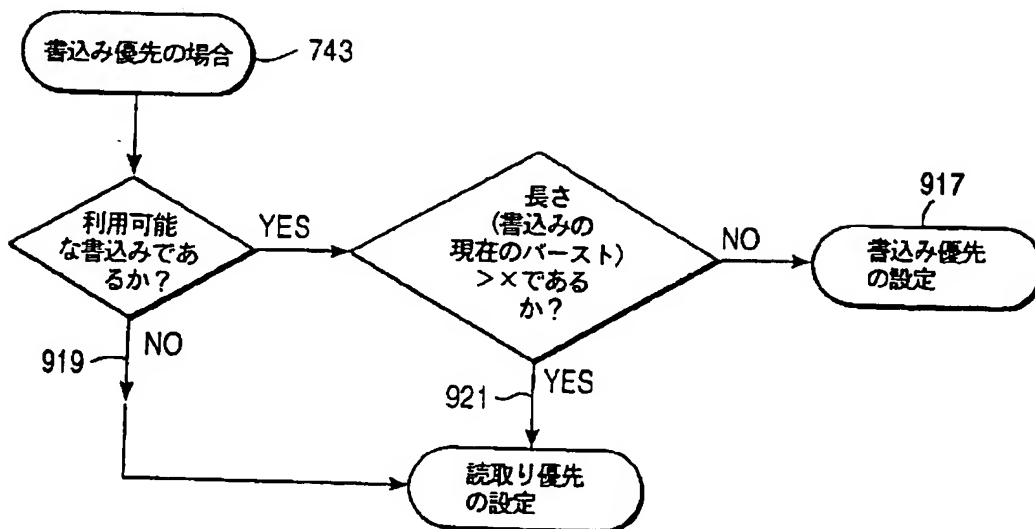
【図13】



【図14】



【図15】



フロントページの続き

(72)発明者 オースティン ジェイ ハイファー
 アメリカ合衆国 マサチューセッツ州
 02160-1706 ニュートン セントラル
 アベニュー 3

THIS PAGE BLANK (USPTO)